2007-08

On Guidelines for Safe Route Redistributions

Le, F.

On Guidelines for Safe Route Redistributions

Franck Le
Carnegie Mellon University
franckle@cmu.edu

Geoffrey G. Xie
Naval Postgraduate School
xie@nps.edu

ABSTRACT

Route redistribution (RR) is becoming a critical tool in enterprise network operations. Like BGP, RR is prone to configuration errors, which may result in severe instabilities such as permanent routing loops and oscillations. In response, router vendors have put forth a set of recommendations on how to configure RR. However, the proposed guidelines are mainly derived from anecdotal experience and based on a limited range of parameters. Having not been subjected to systematic validation, their general effectiveness for preventing routing instabilities is largely unknown. This paper shows that the vendor recommendations do not completely eliminate routing instabilities and have severe limitations in terms of domain backup. It then presents a set of new guidelines with provable properties assuring safety, robustness, reachability, and domain backup. Configurations based on these guidelines allow routing domains of a network to safely exchange information and back up each other, thus increasing the robustness of the network against failures.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Management; C.2.2 [**Network Protocols**]: Routing Protocols

Keywords

Router configuration, route redistribution, routing loop, route oscillations

1. INTRODUCTION

Route redistribution is a growing practice for enterprise networks. Company mergers, multi-vendor environments, and business expansions all contribute to the existence of multiple routing domains or routing instances [10] within corporate networks. Each routing instance consists of a group of routers using the same protocol to exchange *reachability* information. By default routing instances do not share routing information. Operators must explicitly configure border routers (e.g., routers D and E in the network shown in Figure 1) to propagate routes across routing protocol boundaries, a process known as route redistribution (RR) [3].

RR originated as a work-around put in by router vendors to address operators' needs. There is no RFC or open standards on its functionality. Besides expanding reachability, RR was also motivated by a need for *domain backup*. A RR configuration provides domain backup if reachability in the network is maintained, as much as the physical topology allows, even though parts of some routing instances experience link or router failures. For example, consider the network depicted in Figure 1. If link B-C fails, the RIP domain is partitioned into two parts without RR, despite the existence of a physical path from C to B via the OSPF domain. Configuring RR at the two border routers can make such cross-domain backup paths available to routers in both domains.

Border Gateway Protocol (BGP) provides an alternative to RR for integrating the routing instances. RR is often considered preferable to BGP for several reasons. First, BGP requires a full mesh and therefore, every router (including the internal ones) needs to run BGP. Instead, RR only requires configuration at the border routers. One could redistribute the BGP routes into the IGP to avoid the full-mesh, but such an approach still relies on RR. Second, studies have exposed routing instabilities that can result from conflicting BGP policies [6]. Even though [5] showed that existing commercial relationships between Autonomous Systems (customer, provider, peer) impose a partial order on the routes and prevent these instabilities, such relationships do not exist among the routing instances of many enterprise networks. Finally, BGP does not support domain backup in the event of network partitions.

However, RR can be as hard to configure correctly as BGP, as will be explained in Section 2. Misconfigurations of RR can easily result in severe instabilities, including persistent routing loops and permanent route oscillations [3], [2], [9]. Vendors try to mitigate the problems by publishing templates for configuring RR and pointing out common pitfalls of RR configurations through simple examples.

In this paper, we first show that these templates have flaws and the examples have limitations. Then, inspired by the work presented in [5] toward developing guidelines for establishing BGP peering relationships, and building upon

INM'07, August 27–31, 2007, Kyoto, Japan.

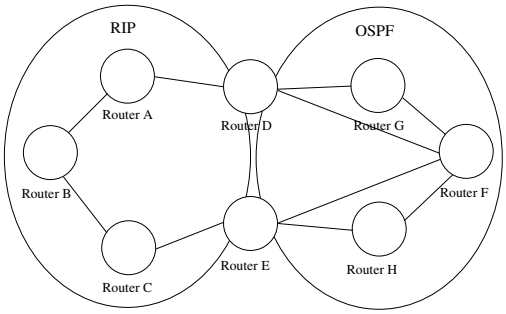

Figure 1: Example enterprise network consisting of two routing domains or routing instances. By default, routers in the RIP domain do not have visibility of routers in the OSPF domain and vice-versa.

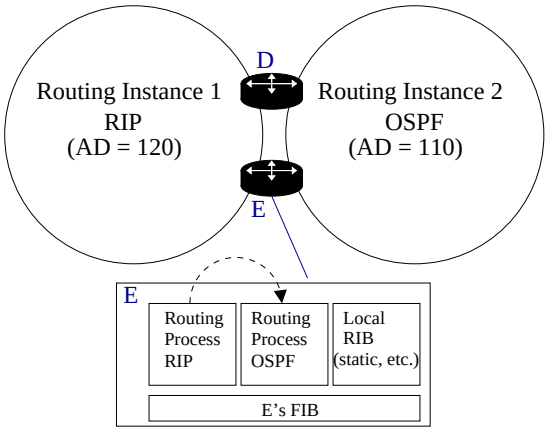

Figure 2: Routing instances and routing processes.

our prior work which developed the first formal model for evaluating the safety of RR configurations [9], we establish guidelines or conditions for configuring RR so that the configuration crafted following the guidelines is free of instabilities and satisfies the reachability and domain backup objectives.

2. FUNDAMENTAL CONCEPTS OF RR

This section presents a brief overview of the most relevant aspects of RR configuration and introduces some of the notations we use. More detailed background information can be found in [9].

A router running multiple routing protocols instantiates a separate *routing process* for each protocol. For example, the network in Figure 1 comprises two routing instances, and as illustrated in Figure 2, routers D and E each runs two routing processes: one for RIP and the other for OSPF. In the rest of the paper, we identify a router by a capital letter and a routing instance by an integer id that is unique within a network. We use $<$*router id*$>.<$*routing instance id*$>$ to name a routing process, e.g., $E.2$ for the OSPF routing process on router E in the example network.

A router running multiple processes may receive several routes to one destination prefix at the same time and need to select one of them to install in its Forwarding Information Base (FIB). To add flexibility to the selection logic, router vendors have introduced a configurable integer parameter per routing process, called *administrative distance* (AD), to aid the ranking of multiple routes to the same destination [4]. Specifically, a route offered by a routing process inherits the AD value of the routing process and the route with the *lowest* AD value is selected and added to the FIB. When not explicitly configured, the AD of a routing process defaults to a protocol-specific value: e.g, 110 for OSPF and 120 for RIP as shown in Figure 2. Likewise, each router's built-in Local RIB (Route Information Base) has default AD values of 0 and 1 respectively for the connected subnets and static routes.

RR can be used to propagate routes across routing processes on the same router. For example, a RR is configured to enable $E.1$ to export routes to $E.2$ in the example network, as illustrated by an arrowed dash line in Figure 2. It is important to note two basic operational characteristics of RR. First, a RR configuration does not mean exporting all routes from the source process (e.g., $E.1$) to the target process (e.g., $E.2$); only those routes that are installed in the FIB, which we term *active* routes of the source process, are eligible for redistribution. Second, RR does not directly impact route selection at the local router; a RR target process will not offer imported routes to the FIB. For example, we assume a route redistributed from $E.1$ into $E.2$. Even though $E.2$ has a lower AD value (110 vs. 120), the route from $E.1$ continues to be the active one in the FIB. However, since it is possible (and quite often) that a RR target process (e.g., $E.2$) disseminates imported routes to other routers (e.g., H) in the same routing instance, the combination of multiple RR configurations, while each is local to a particular router, can have a profound impact on how FIBs are populated across the network.

The interplay between the route selection logic working with routing processes and the RR logic with respect to routing instances introduces tremendous modeling complexity for inferring network wide FIB contents from given per-router RR configurations. In [9], we show that even assuming that the RR configurations lead to convergence, determining if the final routing state is cycle free is NP-hard. On one hand, we observe that a lot of the modeling complexity is due to the total freedom of choosing AD values permitted by the current practice. There are even configuration options to override, on a per destination prefix basis, the AD value of a route learned within the same routing instance. For example, while the AD of $E.2$ is 110, a route received from a peer OSPF process of $E.2$ may be configured to have an AD value different from 110, say 90. On the other hand, we realize it may be possible to take advantage of such flexibility and create guidelines on AD assignment to eliminate routing instabilities.

The following sections discuss guidelines for setting AD values so the resulting set of RR configurations is safe while meeting robustness, reachability and domain backup requirements. We say that a set of RR configurations is *safe* when it always converges to a cycle-free state. We say a set of RR configuration is *robust* if the redistributions remain safe in the event of network failures. Without loss of generality, all discussions are with respect to a single destination prefix, denoted by P, unless noted otherwise.

3. ANALYSIS OF VENDOR DOCUMENTS

Vendors have proposed a number of templates and methods to help operators configure route redistribution [3], [2].

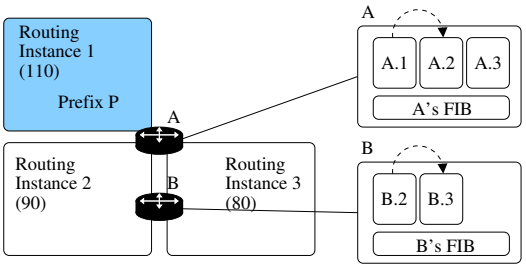

Figure 3: Example of a network where vendors' guidelines do not guarantee safety. A prefix P is originated by instance 1. Router A runs three processes and redistributes from $A.1$ into $A.2$. Router B redistributes from $B.2$ into $B.3$.

Although the methods differ on the implementation means (e.g., some rely on access-lists while others utilize tags or keywords), they all achieve the same goal stipulated by the following vendors' guideline [3]:

Vendors Guideline: *Do not redistribute a route originally received from a routing process back into that routing process.*

This guideline is intended to prevent the formation of routing loops and route oscillations. To illustrate its rationale, we assume a prefix originated by routing instance 1 in Figure 2. If E redistributes from $E.1$ into $E.2$ and D redistributes from $D.2$ into $D.1$, a routing loop may arise between routing instances 1 and 2. Also, if both redistributions from 1 into 2 and from 2 into 1 are enabled at D and E, permanent route oscillations can happen [2].

However, this principle is not sufficient to guarantee safety. Configurations compliant with this suggested principle can still be vulnerable to instabilities. Figure 3 provides such an example. The depicted configuration complies with the vendors' guideline since routes to P are redistributed only from 1 into 2 (at A), and from 2 into 3 (at B). However, as illustrated in Figure 4, such configuration results in permanent route oscillations. We implemented a similar topology and observed a permanent route flap. To represent the propagation of the routes in the network and the sequence of routing states observed, we introduce the following notion of time based on variable **t**. We initialize **t** to 1 to at the initial state, and increment **t** by one whenever the system transitions to a different state. In addition, we color a routing instance in white when it does not have a route to the destination, and in blue shade when it does. The redistributions are represented by arrows between the routing processes. An arrow is dashed when the redistribution is configured but not active. An arrow is solid when the redistribution is active – i.e., the source process's route to P is selected by the FIB. Finally, at each router, the selected routing process is represented by a double line border (e.g., at t=2, $A.1$ is the selected process for destination P at router A).

t=1 We assume a prefix P originated by instance 1.

t=2 A learns a route through $A.1$ and redistributes the route from $A.1$ into $A.2$.

t=3 B receives the route through $B.2$, installs it in its FIB and redistributes the route from $B.2$ into $B.3$.

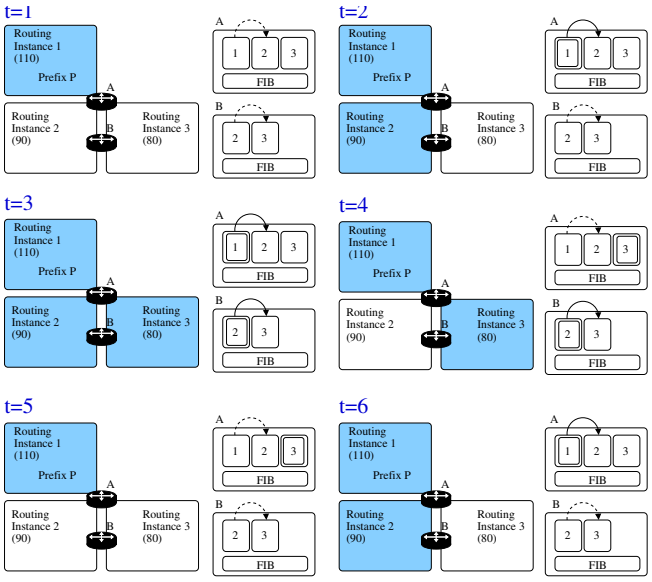

Figure 4: Illustration of a permanent route flap in the network of Figure 3

t=4 A receives two routes to the destination: one from $A.1$ and another one from $A.3$. Because $A.3$ has a lower AD, $A.3$ becomes the selected routing process. Consequently, A stops redistributing from $A.1$ into $A.2$.

t=5 Because A stopped redistributing from $A.1$ into $A.2$, B no longer receives any announcement. B removes the route and stops announcing it into $B.3$.

t=6 Consequently, $A.3$ no longer receives a route to the destination either. We note that this state is identical to the one at t=2, and consequently we have a permanent route oscillation.

In addition to the lack of safety guarantees, because the guideline prevents the re-injection of a route into a routing instance that initially announced the route, none of the existing methods supports domain backup in the event of network partitions. To illustrate the issue, we consider the topology in Figure 1. We assume that the link B-C fails. As a consequence, the RIP domain is divided into two partitions ($\{A, B, D\}$ and $\{C, E\}$). While there exists a physical path for C to reach B (e.g., through path C-E-F-D-A-B), current methods do not allow it. It is important to note that some of the vendor solutions claim to provide domain backup. However, the vendors' definition of "domain backup" has a narrower scope than ours and only focuses on the border processes. For example, in the event of the failure of the link B-C, some of the solutions allow E to still have a route to B – but none of the existing methods allow internal routers (e.g., C) to learn a backup route upon a network failure.

Finally, because [3], [2] mainly provide templates instead of guidelines, the solutions are difficult to generalize and to apply to networks whose topology does not exactly match the ones from the provided examples. Large operational network often comprises more than two routing instances [10], and in those cases, it is unclear what redistributions should be enabled and what AD values should be assigned to the different processes.

4. PROPOSED GUIDELINES

In this section, we present five new guidelines for configuring route redistribution. We present them according to the properties they provide:

- *Safety:* The network always converges to an acyclic routing state.
- *Robustness:* The route redistributions remain safe in the event of network failures.
- *Reachability:* The network always converges to a routing state where each routing instance that needs to have a route to the destination has a loop-free and stable route.
- *Domain backup:* Reachability is guaranteed despite link failures, router failures, and network partitions, as long as a physical path to the destination exists.

4.1 Guideline for safety

Guideline 1 (G-1): *For a route that is redistributed through a sequence of routing instances, its AD value should (i) increase every time the route is redistributed, and (ii) not decrease when propagated within a routing instance.*

Proposition 4.1: Guideline 1 (G-1) guarantees safety.

PROOF SKETCH. Route redistribution follows a distance-vector protocol behavior [9]: when a routing instance u redistributes a prefix P, it only announces the presence (or withdrawal) of a route to its direct neighbors (for which RR is enabled). However, u does not announce the changes related to P to all the routing instances in the network. Consequently, the routing instances do not have a global view of the network but only know the next-hop where to forward the traffic.

[7] introduced a formal framework to study routing protocols, including compositions of routing protocols where for example, an algebra is used between administrative entities and a different algebra is used inside of each administrative entity.

Because RR is a distance-vector protocol used between routing instances, the application of the metarouting framework [7] highlights that strict monotonicity (SM) is a sufficient condition guaranteeing the correctness of the protocol. Strict monotonicity means that the preference of a path strictly decreases as it is extended [11].

We use the AD associated with each redistributed prefix to reflect the preference of a route. The AD can well serve our purposes since in the route selection procedure, the AD is the first considered criteria and routes with higher AD are less preferred. Condition (i) ensures that the preferrence of a route is decreased every time it is redistributed to a routing instance. However, contrary to traditional distance-vector protocols where the preference is attached to the route (e.g., through the "metric" field in RIP messages), the AD is local to a router and not propagated in the signaling messages. Condition (ii) addresses this limitation and makes sure that the preference of route does not increase when it propagates within a routing instance. These two conditions ensure that the preference of the route keeps decreasing as it is redistributed between routing instances and as such, guarantees the safety of the redistributions. □

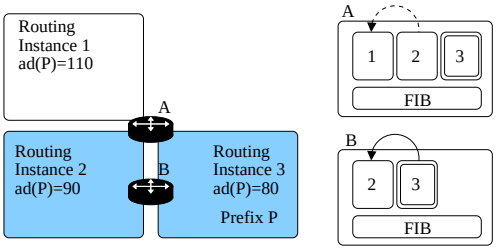

Figure 5: The configuration is robust but does not guarantee reachability. Routing instance 1 does not receive a route to the destination

4.2 Guideline for robustness

Guideline 2 (G-2): *Configure one unique AD value for prefix P in all routing processes within each routing instance. Only redistribute a route to P to a routing instance where the prefix is configured with a higher AD.*

Proposition 4.2: Guideline 2 (G-2) guarantees robustness.

PROOF SKETCH. A configuration compliant with Guideline 2 (G-2) is always compliant with Guideline 1 (G-1) independently of network failures: Since routes are only redistributed to routing processes with higher AD values, condition (i) is always satisfied, and because all routing processes of each routing instance present a same AD value, condition (ii) is also always satisfied.

As such, even in the event of network failures, a configuration compliant with G-2 remains safe. □

4.3 Guideline for robustness & reachability

A RR configuration may be robust but may not guarantee rechability. To illustrate it, we assume the network from Figure 3 and but this time have prefix P originated by routing instance 3. As depicted in Figure 5, we assume that two redistributions are configured for this prefix: from $B.3$ into $B.2$ and from $A.2$ into $A.1$. Consequently, routing instance 2 learns a route to P through B (which redistributes P from $B.3$ into $B.2$). A, which runs 3 routing processes (1, 2, 3), receives two routes to P (from $A.3$ and $A.2$). Because $A.3$ presents a lower AD values than $A.2$, $A.3$ becomes the selected routing process at A, and A does not redistribute from 2 into 1. As such, routing instance 1 does not have any route to P.

Given a network topology with a set of originating routing instances for prefix P, a set of routing instances requiring reachability to P and a set of redistributing routers, we seek to determine the redistributions to enable and the AD values to assign so that the configuration is robust and in the absence network failures, the routing instances requiring reachability have a stable route to P.

We define a *redistribution path*, (u_0-r_0-u_1-r_1-...-u_n-r_n), as a sequence of routing instances (u_0, u_1, ..., u_n) and routers (r_0, r_1, ..., r_n) such that from each of the routing instances u_i, ($i \in$ [0,n-1]), redistribution at router r_i to the next routing instance in the sequence u_{i+1} is configured. The first routing instance u_0 must be an originating routing instance. For example, in Figure 5, 3-B-2 is a redistribution path.

Guideline 3 (G-3): *In addition to complying with G-2, the*

set of resulting redistribution paths must cover all routing instances that require reachability. A router only redistributes from its routing process with the lowest AD value.

Proposition 4.3: Guideline 3 (G-3) guarantees robustness and reachability in the absence of failures.

PROOF SKETCH. A RR configuration compliant with G-3 is compliant with G-2. As such, a RR configuration compliant with G-3 is robust.

The additional conditions provide reachability. First, according to the guideline, the redistribution paths cover all routing instances that require reachability. Consequently, if each redistribution is active, every routing instance requiring reachability does indeed receive a route. We show that in the absence of failures, every redistribution is active: Since every redistributing router r is configured to redistribute from its routing instance u with the lowest AD, if u has a route, $r.u$ is the selected routing process at r and the redistributions from $r.u$ are active. As the route propagates from the originating vertex along the selected redistribution paths, when a routing instance learns a route, the redistributions from that routing instance become active. As such, all redistribution paths gradually become active. □

For the network in Figure 5 to satisfy G-3, one solution consists in redistributing not from $A.2$ into $A.1$ but from $A.3$ into $A.1$. In that case, the redistributions comply with G-3. Such modification allows every routing instance to have a route to P. B redistributes from $B.3$ into $B.2$, and A redistributes from $A.3$ into $A.1$. All three routing instances 1, 2 and 3 have a route to P.

4.4 Guidelines for robustness, reachability & domain backup

This section addresses the configuration of RR such that the redistributions are robust, guarantee reachability and provide domain backup (i.e., protect against failures such as link failure, router failure, network partition, etc.).

First, we introduce the deinition of *path disjointness*: two redistribution paths P and Q are *redistributing-router disjoint* if for every redistributing router $r \in P$, $r \notin Q$. Similarly, two redistribution paths P and Q are *routing-instance disjoint* if for every routing instance $u \in P$, $u \notin Q$. Finally, two paths P and Q are *disjoint* if they are redistributing-router disjoint and routing-instance disjoint.

Guideline 4 (G-4): *In addition to complying with G-3, configure multiple disjoint redistribution paths to the routing instances requiring reachability.*

Proposition 4.4: Guideline 4 (G-4) guarantees robustness and reachability even in the event of a failure of a redistributing router or a intermediate routing instance.

PROOF SKETCH. Because complying with G-3, G-4 implies robustness. In addition, with multiple disjoint redistribution paths, a routing instance can still receive a route through the secondary redistribution path when the primary one fails. Redistributing-router disjointness guarantees reachability in the event of failure of a redistributing router, and routing-instance disjointness guarantees reachability in the event of failure of a intermediate routing instance. □

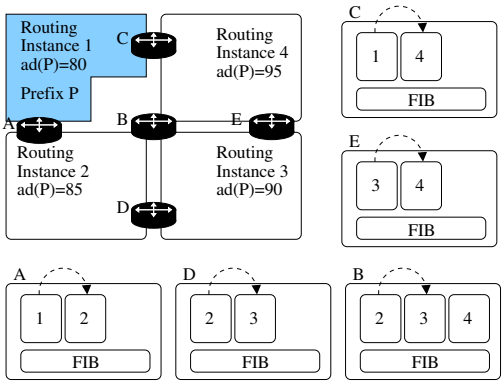

Figure 6: Example of an enterprise network consisting of 4 routing instances and 5 redistributing routers. A prefix P is originated by routing instance 1, and all other routing instances are to have a route to P.

Algorithms from network flow can be applied for identifying the redistribution paths [1].

To protect against the failure of a redistributing router, edge-disjoint pair algorithms can help identifying two redistributing router disjoint paths. To illustrate it, we consider the network in Figure 6. We assume that routing instance 3 is critical. Even though the configuration is compliant with G-3, the failure of A will cause the loss of route at 3. Edge-disjoint pair algorithms can instead highlight two paths (C-E and A-B) such that if a redistributing router (e.g., A, B, C or E) fails, routing instance 3 still gets a route through the secondary path. In that case, E should redistribute not from 3 to 4 but from 4 to 3, and the AD values need to be adjusted to be compliant with guideline G-3 (see Figure 7).

To protect against network partitions, node-disjoint pair algorithms can identify two routing-instance disjoint paths. For example, the partition of routing instance 2 (in Figure 6) into two parts with A on one side and B and D on the other side, prevents routing instance 3 from receiving a route. Node-disjoint pair algorithms can identify two redistributions that are resistant to such failures. Redistributing paths (1, 2, 3) and (1, 4, 3) provide reachability despite the partition of either routing instance 2 or 4.

In the absence of enough disjointness, additional routers or routing instances can be added.

It is important to note that G-4 cannot always guarantee reachability to a routing instance if this same domain gets partitioned. Considering Figure 6, if the originating instance 1 gets partitioned (with A on one side, and C on the other), part of the domain may no longer have a route.

To address this limitation, we propose another method that relies on a functionality implemented by some vendors. Specifically, the method relies on the ability to set the AD value of a route based on the tag a route is carrying, i.e., the implementation should be able to specify: "**if** match tag T, **then** AD = k" [8].

When such capability is supported, the following guideline can be implemented.

Guideline 5 (G-5): *Identify a set of redistribution paths which cover all routing instances requiring reachability. En-*

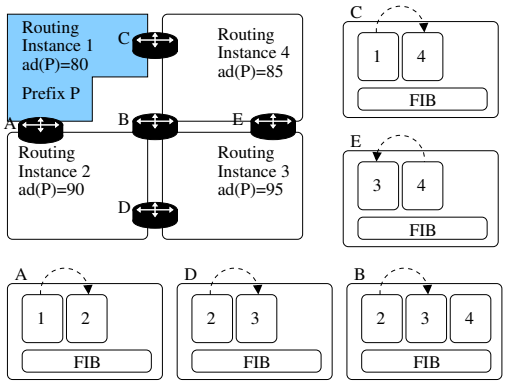

Figure 7: Protection against failures. The redistributing-router disjoint paths from 1 to 3 allows 3 to still receive a route when A fails. 3 still receives a route from C-E.

able mutual route redistribution between the instances on these redistribution paths. Assign a tag to each route, and increment its value when the route is redistributed. Implement a policy at all routers to set the AD value of a newly received route equal to the route's tag value.

Proposition 4.5: Guideline 5 (G-5) guarantees robustness, reachability and domain backup.

PROOF SKETCH. The tag essentially carries the preference of a route, like the distance metric for RIP. Every time a route is redistributed, its preference is decreased. However, the tag field is currently not interpreted as a route preference. We solve this problem in two steps: first incrementing the tag value when redistributing a route, and then setting the AD value of a newly received route equal to the route's tag value at all routers. The approach enforces strict monotonicity (SM) to route redistribution, a sufficient condition guaranteeing its correctness [7]. Consequently, the redistribution paths do not have to be restricted to pre-selected paths and RR behaves like a traditional routing protocol, dynamically discovering the best available path. As such, G-5 guarantees robustness, reachability and domain backup; the detailed arguments are provided in [9]. □

To illustrate how this approach supports domain backup, we assume that the proposed method is adopted in the network from Figure 7, and routing instance 2 is partitioned into two parts with A in one and B and D in the other. In this case, $B.2$ (respectively, $D.2$) no longer receives a route. Instead, B (respectively, D) only receives a route from $B.3$ (resp., $D.3$) with a tag value of at least 2 since it is redistributed twice (by C and by E). B (resp., D) increments the tag value and redistributes the route from $B.3$ (resp., $D.3$) into $B.2$ (resp., $D.2$). Consequently, D (resp., B) receives two routes: one from routing instance 2 – redistributed by B (resp., D) – and one from routing instance 3. Because the route from 3 presents a lower tag value and hence AD value, D (resp., B) maintains $D.3$ (resp., $B.3$) as its selected routing process. The redistributions converge and both partitions of 2 have a route to the destination.

5. CONCLUSION

Routing instabilities due to misconfigured route redistribution may be a major problem in enterprise networks as suggested by the sheer number of vendor documents on how to avoid such problems. This paper showed that the vendor recommendations do not completely eliminate routing instabilities and have severe limitations in terms of domain backup. It then presented a set of new guidelines with provable properties assuring safety, robustness, reachability, and domain backup.

Several aspects of our guidelines require further research. First, the AD assignment is subject to existing constraints not always driven by RR, e.g., the rule of thumb of preferring internal routes over external ones. It remains to be seen if there exists a set of unified guidelines for AD assignment that factors in all known constraints. Second, our guidelines provide a reasonable starting point for developing tools that can automatically generate safe and effective RR configurations based on design requirements. One of our long term research goals is to build and evaluate such tools.

6. ACKNOWLEDGMENTS

This research was sponsored by the NSF under both NeTS Grant CNS-0520210 and a Graduate Research Fellowship. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

7. REFERENCES

[1] Bhandari. *Survivable networks, algorithms for diverse routing*. Springer, 1999.

[2] Cisco. Ospf redistribution among different ospf processes, January 2006. Available at `http://www.cisco.com/warp/public/104/ospfprocesses.pdf`.

[3] Cisco. Redistributing routing protocols, September 2006. Available at `http://www.cisco.com/warp/public/105/redist.pdf`.

[4] Cisco. What is administrative distance?, March 2006. Available at `http://www.cisco.com/warp/public/105/admin_distance.html`.

[5] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proc. ACM SIGMETRICS*, 2000.

[6] T. Griffin, F. B. Shepherd, and G. T. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 2002.

[7] T. G. Griffin and J. L. Sobrinho. Metarouting. In *Proc. ACM SIGCOMM*, 2005.

[8] JUNOS. Junos internet software policy framework configuration guide 8.1.

[9] F. Le, G. G. Xie, and H. Zhang. Understanding route redistribution. Technical Report CMU-CS-07-122, Carnegie Mellon University, April 2007. Available at `http://www.nps.navy.mil/faculty/xie/papers/tech-rr.pdf`.

[10] D. Maltz, G. Xie, J. Zhan, H. Zhang, G. Hjalmtysson, and A. Greenberg. Routing design in operational networks: A look from the inside. In *Proc. ACM SIGCOMM*, 2004.

[11] J. L. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proc. ACM SIGCOMM*, 2003.